Journal of Discrete Algorithms 8 (2010) 131–142

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

On the complexity of finding gapped motifs

Morris Michael [1], François Nicolas *, Esko Ukkonen

Department of Computer Science, PO Box 68 (Gustaf Hällströmin katu 2b), FIN-00014 University of Helsinki, Finland

ARTICLE INFO

Article history:
Received 26 June 2007
Received in revised form 7 January 2009
Available online 4 February 2010

Keywords:
Gapped pattern
Motif discovery
String matching with don't care symbols
NP-complete
Tandem motifs

ABSTRACT

A gapped pattern is a sequence consisting of regular alphabet symbols and of joker symbols that match any alphabet symbol. The content of a gapped pattern is defined as the number of its non-joker symbols. A gapped motif is a gapped pattern that occurs repeatedly in a string or in a set of strings. The aim of this paper is to study the complexity of several gapped-motif-finding problems. The following three decision problems are shown NP-complete, even if the input alphabet is binary. (i) Given a string T and two integers c and q, decide whether or not there exists a gapped pattern with content c (or more) that occurs in T at q distinct positions (or more). (ii) Given a set of strings $\mathcal{S}$ and an integer c, decide whether or not there exists a gapped pattern with content c that occurs at least once in each string of $\mathcal{S}$. (iii) Given m strings with the same length, and two integers c and q, decide whether or not there exists a gapped pattern with content c that matches at least q input strings. We also present a non-naive quadratic-time algorithm that solves the following optimization problem: given a string T and an integer $q \geqslant 1$, compute a maximum-content gapped pattern Q such that q consecutive copies of Q occur in T.

1. Introduction

Finding and representing patterns of symbols that occur repeatedly in a string (or a set of strings) is a basic problem in word combinatorics as well as in many applications including string matching, data compression [19], biological sequence analysis [6], mining of sequential data [5], etc.

The difficulty of pattern synthesis problems (also called *motif-finding problems*) depends on the class of patterns under consideration as well as on the nature, exact or approximate match, of the repeated occurrences. The easiest case is finding exactly repeated substrings. The problem is to find all the substrings that occur more than once in an input string. The suffix-tree techniques are known to give a full solution, even in linear time [6]: a suffix-tree provides a compact representation of the occurrence locations of all repeated substrings. The problem becomes much more difficult as soon as we allow approximation in the repeated occurrences of the substring pattern. For example, finding a closest substring under Hamming distance for an input set of strings is NP-hard [10]; see also [12] and [10] for various approximate pattern synthesis problems under Hamming distance.

This paper focuses on so-called *gapped motifs*. A gapped motif is a repetition of a sequence pattern that may contain, in addition to the regular alphabet symbols, a number of *joker* symbols that match any alphabet symbol. An occurrence of such a pattern has the alphabet symbols as given in the pattern, separated by any symbols as indicated by the jokers of the pattern. So an exact match is required for the regular alphabet symbols of the pattern, and each gap has a fixed length. This contrasts with the problems of finding motifs with constrained gaps [13,18] or episodes [5].

* Corresponding author.
E-mail addresses: morris.michael@ake-software.de (M. Michael), nicolas@cs.helsinki.fi (F. Nicolas), ukkonen@cs.helsinki.fi (E. Ukkonen).
[1] Currently with a.k.e Software GmbH, Hannover, Germany.

1570-8667/$ – see front matter
doi:10.1016/j.jda.2009.12.001

1.1. Notations

For every real number x, $\lceil x \rceil$ denotes the unique integer satisfying $x \leqslant \lceil x \rceil < x + 1$, and $\lfloor x \rfloor$ denotes the unique integer satisfying $x - 1 < \lfloor x \rfloor \leqslant x$.

1.1.1. Words

For all integers p and q, $[\![p, q]\!]$ denotes the set of all integers n such that $p \leqslant n \leqslant q$: for instance, $[\![3, 8]\!] = \{3, 4, 5, 6, 7, 8\}$. An *alphabet* Σ is a set of *symbols*, also called its *letters*. Alphabet $\{0, 1\}$ is chosen as canonical *binary alphabet*. A *string* (over Σ) is a finite sequence of symbols (drawn from Σ). The set of all strings over Σ is denoted $\Sigma^\star$. String concatenation is denoted multiplicatively. For every string W, the *length* of W is denoted $|W|$. The unique string of length zero is called the *empty string*. For every integer $n \geqslant 0$, Σ^n denotes the set of all n-length strings over Σ. For every index $i \in [\![1, |W|]\!]$, $W[i]$ denotes the ith letter of W: $W = W[1]W[2] \cdots W[|W|]$, $W[1]$ is the first letter of W, and $W[|W|]$ is the last letter of W. A *substring* of W is a string of the form $W[i, j] := W[i]W[i + 1] \cdots W[j]$, where indices i and j satisfy $0 \leqslant i - 1 \leqslant j \leqslant |W|$.

1.1.2. Gapped patterns

Throughout this paper, symbol ? plays the role of a *joker*. A string P such that $P[i]$ may equal ? for some indices $i \in [\![1, |P|]\!]$ is rather called a *gapped pattern*.

Definition 1 *(Content).* The *content* of a gapped pattern P is defined as the number of indices $i \in [\![1, |P|]\!]$ such that $P[i] \neq ?$.

Definition 2 *(Match).* Let S be a string and let P be a gapped pattern. We say that P *matches* S whenever $|P| = |S|$ and for every $i \in [\![1, |P|]\!]$, $P[i] = ?$ or $P[i] = S[i]$.

Definition 3 *(Occurrence).* Let T be a string and let P be gapped pattern. Given an index $p \in [\![1, |T| - |P| + 1]\!]$, we say that P *occurs* in T at position p whenever P matches the substring $T[p, p + |P| - 1]$. The set of all indices $p \in [\![1, |T| - |P| + 1]\!]$ such that P occurs in T at position p is denoted $\mathrm{L}_P(T)$. The *number of occurrences* of P in T, denoted $|T|_P$, is defined as the cardinality of $\mathrm{L}_P(T)$.

According to our convention, the empty gapped pattern occurs in T at position p for every $p \in [\![1, |T| + 1]\!]$. Let $T := 01001101110011111001010001111110$ and $P := 00?1?1$. The content of P equals 4; P matches strings $T[3, 8] = 001101$, $T[11, 16] = T[23, 28] = 001111$ and $T[22, 27] = 000111$; P occurs in T at positions 3, 11, 22 and 23; $\mathrm{L}_P(T) = \{3, 11, 22, 23\}$; $|T|_P = 4$.

Note that for any letter $a \neq ?$, $|T|_a = \#\{i \in [\![1, |T|]\!] : T[i] = a\}$.

1.2. Contribution

The aim of this paper is to study the computational complexity of the four motif-finding problems listed below.

(1) Common Gapped Pattern (CGP): "*Given a finite set of strings $\mathcal{S}$ and a non-negative integer c, is there a gapped pattern with content c occurring at least once in each string of $\mathcal{S}$?*"
(2) Gapped Pattern Matching Most Strings (GPMMS): "*Given m strings $S_1, S_2, \ldots, S_m$ with the same length, and two non-negative integers c and q, is there a gapped pattern with content c, matching S_i for at least q distinct indices $i \in [\![1, m]\!]$?*"
(3) Gapped Motif (GM): "*Given a string T and two non-negative integers c and q, is there a gapped pattern with content c occurring in T at q distinct positions or more?*"
(4) Gapped Tandem Repeat (GTR): "*Given a string T and a positive integer q, compute a gapped pattern with maximum content, among all gapped patterns Q such that Q^q occurs in T.*"

We prove that the decision problems CGP, GPMMS and GM are NP-complete; we present a quadratic-time algorithm for GTR.

1.3. Related works

1.3.1. Problems CGP and GPMMS

The "gapless" version of CGP is known as the Longest Common Substring problem and can be solved in linear time [6].

Besides, both CGP and GPMMS are variants of previously studied motif-finding problems based on Hamming distance. The CGP problem is a variant of the Closest Substring problem: "*Given a finite set of strings $\mathcal{S}$ and two non-negative integers d and n, is there an n-length string S such that each string in $\mathcal{S}$ has at least one n-length substring within Hamming distance d of S?*" The GPMMS problem is a variant of the Close to Most String problem: "*Given m strings with the same length n, and two non-negative integers d and q, is there an n-length string S such that at least q input strings are within Hamming distance d of S?*" Both Closest Substring and Close to Most String are NP-hard [10] but Closest Substring admits a Polynomial Time Approximation Scheme on bounded alphabets [12] while approximating Close to Most String is NP-hard [10].

Table 1
Summary of the problem names and of the reductions.

Short name	Full name	Proved NP-hard in	Reduction from
CGP	Common Gapped Pattern	Theorem 6	CGP1
GPMMS	Gapped Pattern Matching Most Strings	Theorem 14	GPMMS1
GM	Gapped Motif	Theorem 32	GM1
CGP1	Common Unary Gapped Pattern	[14] (see Lemma 4)	Independent Set
GPMMS1	Unary Gapped Pattern Matching Most Strings	Lemma 7	Clique
GM1	Unary Gapped Motif	Lemma 25	GPMMS1

1.3.2. Problem GM

As previously mentioned, the "gapless" version of GM can be solved in linear time using suffix-trees.

A variant of GM where the sought gapped pattern is constrained to be of the form $U?^kV$, k being a positive integer and U and V being gapless strings, has also been considered: an $O(n \log n)$-time algorithm has been designed [4].

For fixed q, the problem of extracting the gapped patterns that occur q times or more in an input string has been addressed in two ways [15,1,16,17,2]. To deal with a potential avalanche of output, a polynomial delay enumeration algorithm has been proposed [2]. The other approach is to compute in polynomial time a small subset of the motifs that somehow represents the whole [15,1,16,17].

1.3.3. Problem GTR

Define a *periodicity* as a gapless string of the form S^qS' where S is a non-empty string, where q is an integer greater than one and where S' is a proper prefix of S. A compact representation of all the periodicities that occur in an input string is computable in linear time [8]. The more practical problem of finding approximate periodicities has also been addressed [11,9]. Note that there is no canonical definition for the notion of approximate periodicities. So far, each proposed definition refers to a metric on strings, which is either Hamming [9] or edit [11] distance. The GTR problem asks for uncovering a gapped periodicity in the input string.

1.4. Organization of the paper

In order to decompose long NP-hardness proofs, the following "binary-unary" versions of CGP, GPMMS and GM are used as auxiliary problems.

(5) Common Unary Gapped Pattern (CGP1): "*Given a finite set of binary strings $\mathcal{S} \subseteq \{0,1\}^\star$ and a non-negative integer c, is there a gapped pattern $P \in \{1,?\}^\star$ such that P has content c, and for each $S \in \mathcal{S}$, P occurs in S?*"
(6) Unary Gapped Pattern Matching Most Strings (GPMMS1): "*Given m strings $S_1, S_2, \ldots, S_m \in \{0,1\}^n$, and two non-negative integers c and q, is there a gapped pattern $P \in \{1,?\}^n$ with content c such that P matches S_i for at least q distinct indices $i \in [\![1,m]\!]$?*"
(7) Unary Gapped Motif (GM1): "*Given a binary string $T \in \{0,1\}^\star$, and two non-negative integers c and q, is there a gapped pattern $Q \in \{1,?\}^\star$ with content c such that Q occurs in T at q distinct positions or more?*"

Problems CGP, GPMMS, GM, CGP1, GPMMS1 and GM1 are trivially in NP since in each case, the sought gapped patterns can be used as certificates for yes-instances. Note in passing that CGP1 (resp. GPMMS1, resp. GM1) is *not* a restriction of CGP (resp. GPMMS, resp. GM).

The four remaining sections of this paper are organized as follows. Decision problems CGP1 and CGP (resp. GPMMS1 and GPMMS, resp. GM1 and GM) are shown NP-hard in Section 2 (resp. Section 3, resp. Section 4). The quadratic algorithm for GTR is presented in Section 5. The organization of the reductions is summarized in Table 1. Note that Sections 3 and 4 are not independent.

2. Complexity of the Common Gapped Pattern problem

In this section we easily deduce the NP-hardness of CGP from previously known hardness results. The following optimization version of CGP1 has been introduced in [14] under the name region specific Maximum Weight Lossless Seed (MWLS): "*Given a finite set of binary strings $\mathcal{S} \subseteq \{0,1\}^\star$, find a gapped pattern $P \in \{1,?\}^\star$ with maximum content such that P occurs in S for every $S \in \mathcal{S}$.*" The study of MWLS yielded:

Lemma 4. *(See [14].) The* CGP1 *problem is* NP*-complete.*

As suggested by the next remark, a little padding suffices to obtain a Karp-reduction from CGP1 to CGP.

Remark 5. Let ℓ be a non-negative integer and let P be a gapped pattern: P occurs in 1^ℓ iff both $P \in \{1,?\}^\star$ and $|P| \leqslant \ell$ hold.

Theorem 6. *The* CGP *problem is* NP*-complete, even if the input alphabet is binary.*

Proof. Consider the function that maps each instance $(\mathcal{S}, c)$ of CGP1 to the instance $(\mathcal{S} \cup \{1^\ell\}, c)$ of CGP, where $\ell := \min_{S\in\mathcal{S}} |S|$. This transformation is a Karp-reduction from CGP1 to CGP by Remark 5. Hence, CGP is NP-hard by Lemma 4. □

Noteworthy is that the result proved in [14] is actually stronger than Lemma 4: MWLS was proven NP-hard to approximate within ratio $(\#\mathcal{S})^{0.5-\epsilon}$ for any real $\epsilon > 0$. The reduction from CGP1 to CGP exhibited in the proof of Theorem 6 yields the same $(\#\mathcal{S})^{0.5-\epsilon}$-approximation lower bound for the content maximization version of CGP.

3. Complexity of the Gapped Pattern Matching Most Strings problem

In this section, we demonstrate that GPMMS is NP-complete by reduction from the well-known graph-theoretic problem Clique, whose definition is recalled below. GPMMS1 is used as an auxiliary problem.

All graphs involved in this paper are simple and undirected. Given a graph G, a *clique* in G is defined as a set of pairwise adjacent vertices of G. The Clique problem is: "*Given a graph G and a non-negative integer k, is there a clique in G with cardinality k?*" The NP-completeness of Clique was proved by Karp in 1972 [7].

Lemma 7. *The* GPMMS1 *problem is* NP*-complete.*

Proof. We reduce Clique to GPMMS1.

Let G be a graph and let k be a non-negative integer. Denote by n and m the number of vertices and the number of edges of G, respectively. Let $v_1, v_2, \ldots, v_n$ be an enumeration of the vertices of G, and let $e_1, e_2, \ldots, e_m$ be an enumeration of the edges of G: G is completely defined by its vertex set $\{v_1, v_2, \ldots, v_n\}$ and its edge set $\{e_1, e_2, \ldots, e_m\}$.

Remark 8. Consider a vertex set $K \subseteq \{v_1, v_2, \ldots, v_n\}$ with cardinality k. Then, K is a clique in G iff there are at least $k(k-1)/2$ distinct indices $i \in [\![1, m]\!]$ such that the two endpoints of edge e_i belong to K.

For every gapped pattern $P \in \{1, ?\}^n$, let K_P be the set of all vertices v_j with $j \in [\![1, n]\!]$ such that $P[j] = ?$. For instance, if $P = ?111??1$ then $K_P = \{v_1, v_5, v_6\}$. In our reduction, each gapped pattern $P \in \{1, ?\}^n$ is thought as a characteristic function of K_P.

Remark 9. The mapping $P \mapsto K_P$ induces a bijection from $\{1, ?\}^n$ to the set of all subsets of the vertex set of G.

Remark 10. For every gapped pattern $P \in \{1, ?\}^n$ with content $n-k$, K_P is of cardinality k.

Now, transform the instance (G, k) of Clique into an instance $((S_1, S_2, \ldots, S_m), c, q)$ of GPMMS1. For every $(i, j) \in [\![1, m]\!] \times [\![1, n]\!]$, define symbol $s_{i,j} \in \{0, 1\}$ as follows:

$$s_{i,j} := \begin{cases} 0 & \text{if } v_j \text{ is an endpoint of } e_i, \\ 1 & \text{otherwise.} \end{cases}$$

For each $i \in [\![1, m]\!]$, compute the n-length binary string $S_i := s_{i,1}s_{i,2}\cdots s_{i,n}$, and let $c := n - k$ and $q := k(k-1)/2$. For instance, if $n = 7$ and if e_3 links v_3 and v_6 then $S_3 = 1101101$. Clearly, instance $((S_1, S_2, \ldots, S_m), c, q)$ of GPMMS1 is computable from (G, k) in polynomial time. It remains to prove the next claim.

Claim 11. *(G, k) is a yes-instance of* Clique *if, and only if, $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of* GPMMS1.

Remark 12. For every gapped pattern $P \in \{1, ?\}^n$ and every index $i \in [\![1, m]\!]$, P matches S_i iff the two endpoints of edge e_i belong to K_P.

With this remark in hand, we turn to the proof of Claim 11.

(*if*) Assume that $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1. This means that there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that the index set $I := \{i \in [\![1, m]\!]:\ P \text{ matches } S_i\}$ has cardinality at least $q = k(k-1)/2$. Then, vertex set K_P is of cardinality k by Remark 10, and for each $i \in I$, the two endpoints of edge e_i belong to K_P by Remark 12. Therefore, K_P is a clique in G according to Remark 8, and (G, k) is a yes-instance of Clique.

(*only if*) Conversely, assume that (G, k) is a yes-instance of Clique. Then, there exists a clique K with cardinality k in G. According to Remark 9, there also exists $P \in \{1, ?\}^n$ such that $K = K_P$. Gapped pattern P has content $n-k = c$ by Remark 10. Moreover, for each index $i \in [\![1, m]\!]$ such that the two endpoints of edge e_i belong to K, P matches S_i by Remark 12. Since K is a clique in G with cardinality k, there are $k(k-1)/2 = q$ such indices i (see Remark 8). Therefore, $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1. □

Building on the next remark, GPMMS1 Karp-reduces to GPMMS *via* a padding argument.

Remark 13. Let n be a non-negative integer and let P be a gapped pattern: P matches 1^n iff $P \in \{1, ?\}^n$.

Theorem 14. *The* GPMMS *problem is* NP*-complete, even if the input alphabet is binary.*

Proof. We reduce GPMMS1 to GPMMS in order to apply Lemma 7.

Let $((S_1, S_2, \ldots, S_m), c, q)$ be an instance of GPMMS1. Recall that n denotes the non-negative integer such that $S_i \in \{0,1\}^n$ for every $i \in [\![1, m]\!]$. Transform $((S_1, S_2, \ldots, S_m), c, q)$ into the instance $((S'_1, S'_2, \ldots, S'_{m'}), c, q')$ of GPMMS where $m' := 2m + 1$, $S'_i := S_i$ for every $i \in [\![1, m]\!]$, $S'_i := 1^n$ for every $i \in [\![m + 1, m']\!]$, and $q' = q + m + 1$. We claim that this transformation is a Karp-reduction. Clearly, it is computable in polynomial time. Hence, it remains to check the next claim.

Claim 15. *$((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of* GPMMS1 *if, and only if, $((S'_1, S'_2, \ldots, S'_{m'}), c, q')$ is a yes-instance of* GPMMS.

For every gapped pattern P with length n, let $I_P := \{i \in [\![1, m]\!]\colon\ P \text{ matches } S_i\}$ and $I'_P := \{i \in [\![1, m']\!]\colon\ P \text{ matches } S'_i\}$.

Lemma 16. *For every $P \in \{1, ?\}^n$, $\#I'_P = \#I_P + m + 1$.*

Proof. For any gapped pattern P with length n, the inclusion $I'_P \subseteq I_P \cup [\![m+1, m']\!]$ is clear since $S'_i = S_i$ for every $i \in [\![1, m]\!]$. Moreover, equality $I'_P = I_P \cup [\![m + 1, m']\!]$ holds whenever $P \in \{1, ?\}^n$ since $S'_{m+1} = S'_{m+2} = \cdots = S'_{m'} = 1^n$ (see Remark 13). Therefore, the rule of sum yields

$$\#I'_P = \#I_P + \#[\![m + 1, m']\!] = \#I_P + m' - m = \#I_P + m + 1.$$

This concludes the proof of Lemma 16. □

With Lemma 16 in hand, let us turn to the proof of Claim 15.

(*only if*) Assume that $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1. This means that there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that the cardinality of I_P is at least q. Then, according to Lemma 16, the cardinality of I'_P is at least $q + m + 1 = q'$, and thus $((S'_1, S'_2, \ldots, S'_{m'}), c, q')$ is a yes-instance of GPMMS.

(*if*) Conversely, assume that $((S'_1, S'_2, \ldots, S'_{m'}), c, q')$ is a yes-instance of GPMMS. This means that there exists a gapped pattern P with content c such that the cardinality of I'_P is at least q'. In particular, the cardinality of I'_P is greater than m. Hence, there exists an element $i'_0 \in I'_P$ with $i'_0 \notin [\![1, m]\!]$: P matches $S'_{i'_0} = 1^n$, and thus P belongs to $\{1, ?\}^n$ by Remark 13. Furthermore, Lemma 16 yields $\#I_P = \#I'_P - (m + 1) \geqslant q' - (m + 1) = q$. Therefore, $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1. □

4. Complexity of the Gapped Motif problem

In this section we prove that the GM problem is NP-complete by reduction from GPMMS1, using GM1 as auxiliary problem.

4.1. Reduction *from* GPMMS1 *to* GM1

The gadget put to use in the reduction is built on the basis of Golomb rulers [3].

A *Golomb ruler* is an integer set Γ satisfying the following property: for every integer $m \geqslant 1$, there exists at most one ordered pair $(\alpha, \beta) \in \Gamma \times \Gamma$ such that $\beta - \alpha = m$. Integers belonging to a Golomb ruler are called its *marks*. Less formally, a Golomb ruler is a ruler such that no two pairs of distinct marks measure the same distance. For instance, $\{0, 1, 4, 9, 11\}$ is a 5-mark Golomb ruler.

Lemma 17. *(See [14].) For every integer $n \geqslant 0$, the integer set $\Gamma_n := \{(j-1)n^2 + j^2\colon\ j \in [\![1, n]\!]\}$ is an n-mark Golomb ruler.*

Proof. It suffices to check that, for any ordered pair of indices (i, j) with $1 \leqslant i < j \leqslant n$, (i, j) can be written as a function of the difference

$$m := \big((j-1)n^2 + j^2\big) - \big((i-1)n^2 + i^2\big).$$

More precisely, we show the following two equalities:

$$i = \frac{1}{2}\left(\frac{m \bmod n^2}{\lfloor m/n^2 \rfloor} - \lfloor m/n^2 \rfloor\right) \quad \text{and} \quad j = \frac{1}{2}\left(\frac{m \bmod n^2}{\lfloor m/n^2 \rfloor} + \lfloor m/n^2 \rfloor\right). \tag{1}$$

Table 2
The string $\widehat{S}$ for each $S \in \{0^n, 1111, 1100, 11111, 00110\}$.

$S =$	0^n	1111	1100	11111	00110
$\widehat{S} =$	0^{n^3}	$10^{18}10^{20}10^{22}1$	$10^{18}10^{44}$	$10^{27}10^{29}10^{31}10^{33}1$	$0^{58}10^{31}10^{34}$

Set $q := j - i$ and $r := j^2 - i^2$. It is clear that both i and j can be written as functions of q and r:

$$i = \frac{1}{2}\left(\frac{r}{q} - q\right) \quad \text{and} \quad j = \frac{1}{2}\left(\frac{r}{q} + q\right). \tag{2}$$

Furthermore, q and r clearly satisfy $m = qn^2 + r$ and $0 \leqslant r \leqslant n^2 - 1$. Hence, q and r are respectively the quotient and the remainder of the division of m by n^2:

$$q = \lfloor m/n^2 \rfloor \quad \text{and} \quad r = m \bmod n^2. \tag{3}$$

Combining (2) and (3) yields (1). □

For instance $\Gamma_0 = \emptyset$, $\Gamma_1 = \{1\}$, $\Gamma_2 = \{1, 8\}$, $\Gamma_3 = \{1, 13, 27\}$, $\Gamma_4 = \{1, 20, 41, 64\}$ and $\Gamma_5 = \{1, 29, 59, 91, 125\}$. Note that for every positive integer n, Γ_n is a subset of $[\![1, n^3]\!]$ containing 1 and n^3 as elements. Moreover, Γ_n is computable from n in a time polynomial in n.

Definition 18. For every $S \in \{0, 1\}^\star$, define $\widehat{S}$ as the binary string with length $|S|^3$ given by:

- $\widehat{S}[(j-1)|S|^2 + j^2] := S[j]$ for every $j \in [\![1, |S|]\!]$, and
- $\widehat{S}[k] := 0$ for every $k \in [\![1, |S|^3]\!] \setminus \Gamma_{|S|}$.

See Table 2 for some examples.

Lemma 19. *Let S be a binary string and let W be a gapped pattern such that $1W1$ occurs in $\widehat{S}$. There is a single occurrence of $1W1$ in $\widehat{S}$ and its position is completely determined by $|S|$ and $|W|$.*

Proof. Let $p \in \mathrm{L}_{1W1}(\widehat{S})$. The first letter and the last letter of $1W1$ occur in $\widehat{S}$ at positions p and $p + |W| + 1$, respectively. Then, $\widehat{S}[p] = 1$ and $\widehat{S}[p + |W| + 1] = 1$ require $p \in \Gamma_{|S|}$ and $p + |W| + 1 \in \Gamma_{|S|}$, respectively. Since in addition $\Gamma_{|S|}$ is a Golomb ruler (Lemma 17), p is completely determined by $|S|$ and $|W|$. □

Lemma 20. *Let n and c be two integers such that $n \geqslant 0$ and $c \geqslant 2$, and let $\mathcal{S}$ be a non-empty subset of $\{0, 1\}^n$. The following two assertions are equivalent*:

(i) *there exists a gapped pattern $Q \in \{1, ?\}^\star$ with content c such that Q occurs in $\widehat{S}$ for every $S \in \mathcal{S}$, and*
(ii) *there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that P matches S for every $S \in \mathcal{S}$.*

Proof. (i) ⇒ (ii). First, assume that assertion (i) holds. This means that there exists a gapped pattern $Q \in \{1, ?\}^\star$ with content c such that Q occurs in $\widehat{S}$ for every $S \in \mathcal{S}$. Without loss of generality, we may delete the leading and trailing ? symbols of Q: now, the first and last letters of Q are 1s. Since in addition, the content of Q is greater than one, Q is not reduced to a single letter 1. Hence, Q is of the form $Q = 1W1$ for some $W \in \{1, ?\}^\star$, and thus Lemma 19 applies in the following way: for each $S \in \mathcal{S}$, Q occurs in $\widehat{S}$ at some position p which is independent of S. Let $Q' := ?^{p-1}\,Q\,?^{n^3-|Q|-p+1}$: Q' belongs to $\{1, ?\}^{n^3}$, and for each $S \in \mathcal{S}$, Q' matches $\widehat{S}$. Then, define $P \in \{1, ?\}^n$ by: for each $j \in [\![1, n]\!]$, $P[j] := Q'[(j-1)n^2 + j^2]$. It is easy to see that P matches S for every $S \in \mathcal{S}$. Moreover, for any $S \in \mathcal{S}$ and any $k \in [\![1, n^3]\!] \setminus \Gamma_n$, $\widehat{S}[k] = 0$ requires $Q'[k] = ?$. It follows $\mathrm{L}_1(Q') \subseteq \Gamma_n$ so P has content c: each occurrence of letter 1 in Q' corresponds to an occurrence of 1 in P. Hence, assertion (ii) holds.

(ii) ⇒ (i). Conversely, assume that assertion (ii) holds. This means that there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that P matches S for every $S \in \mathcal{S}$. Let $Q \in \{1, ?\}^{n^3}$ be the gapped pattern defined by:

- $Q[(j-1)n^2 + j^2] := P[j]$ for every $j \in [\![1, n]\!]$, and
- $Q[k] := ?$ for every $k \in [\![1, n^3]\!] \setminus \Gamma_n$.

Less formally, Q is built from P in the same way as $\widehat{S}$ is built from S, except that joker symbol ? plays the role of letter 0. It is easy to see that Q has content c and that Q matches $\widehat{S}$ for every $S \in \mathcal{S}$. Hence, assertion (i) holds. □

Remark 21. Let n and N be two integers with $N > 0$. There exists a single integer ρ such that $n \in [\![(\rho-1)N+1, \rho N]\!]$, and this integer equals $\lceil n/N \rceil$.

In the same way, there exists a single integer ρ such that $n \in [\![\rho N, (\rho+1)N-1]\!]$ and this integer equals $\lfloor n/N \rfloor$.

Lemma 22. *Let M, N, p be three positive integers. Let $U_1, U_2, \ldots, U_M$ be M strings with the same length N, let W be a non-empty (gapless) string that occurs in the concatenation $U_1U_2\cdots U_M$ at position p, let $\alpha := \lceil p/N \rceil$, and let $\beta := \lceil (p+|W|-1)/N \rceil$.*

(i) *The first letter and the last letter of W occur in U_α and U_β, respectively.*
(ii) *W occurs in the string $U_\alpha U_{\alpha+1}\cdots U_\beta$ at position $p-(\alpha-1)N$.*
(iii) *$\beta-\alpha$ equals either $\lfloor (|W|-1)/N \rfloor$ or $\lceil (|W|-1)/N \rceil$.*

Proof. The proofs of points (i) and (ii) of Lemma 22 rely on the following trivial remark.

Remark 23. Let T be a string and let i_1, i_2, j_1, j_2 be four indices satisfying $1 \leqslant j_1 \leqslant i_1 \leqslant i_2 \leqslant j_2 \leqslant |T|$. Then $T[i_1, i_2]$ occurs in $T[j_1, j_2]$ at position $i_1 - j_1 + 1$.

Throughout the rest of the proof, T denotes the string $U_1U_2\cdots U_M$.

Proof of point (i). Apply Remark 21 with $n = p$. This yields:

$$(\alpha-1)N+1 \leqslant p \leqslant \alpha N, \tag{4}$$

and thus Remark 23 applies with

$$i_1 = i_2 = p, \qquad j_1 = (\alpha-1)N+1, \qquad j_2 = \alpha N.$$

It follows that $W[1] = T[p]$ occurs in $T[(\alpha-1)N+1, \alpha N] = U_\alpha$.

In the same way, apply Remark 21 with $n = p+|W|-1$ to obtain inequalities

$$(\beta-1)N+1 \leqslant p+|W|-1 \leqslant \beta N, \tag{5}$$

and then apply Remark 23 with

$$i_1 = i_2 = p+|W|-1, \qquad j_1 = (\beta-1)N+1, \qquad j_2 = \beta N.$$

We obtain that $W[|W|] = T[p+|W|-1]$ occurs in $T[(\beta-1)N+1, \beta N] = U_\beta$.

We have thus shown point (i).

Proof of point (ii). Eqs. (4) and (5) yield

$$(\alpha-1)N+1 \leqslant p \leqslant p+|W|-1 \leqslant \beta N,$$

and thus Remark 23 applies with

$$i_1 := p, \qquad i_2 := p+|W|-1, \qquad j_1 := (\alpha-1)N+1, \qquad j_2 := \beta N.$$

Hence, $W = T[p, p+|W|-1]$ occurs in $T[(\alpha-1)N+1, \beta N] = U_\alpha U_{\alpha+1}\cdots U_\beta$ at position $i_1 - j_1 + 1 = p-(\alpha-1)N$.

We have thus shown point (ii).

Proof of point (iii). It is easy to see that, for any two real numbers x and y, $\lceil y \rceil - \lceil x \rceil$ equals either $\lfloor y-x \rfloor$ or $\lceil y-x \rceil$. Picking $x := p/N$ and $y := (p+|W|-1)/N$ proves point (iii). □

According to points (i) and (ii) of Lemma 22, $\beta-\alpha+1$ equals the number of blocks U_i (with $i \in [\![1, M]\!]$) being overlapped by the occurrence of W in $U_1U_2\cdots U_M$ at position p. However, $\lfloor (|W|-1)/N \rfloor$ and $\lceil (|W|-1)/N \rceil$ may be two distinct integers, and thus the length of W may not completely determine $\beta-\alpha+1$. To illustrate point (iii) of Lemma 22, consider the case where $N := 4$, $p_1 := 2$, $p_2 := 8$, $U_1 := \texttt{xabb}$, $U_2 := \texttt{baba}$, $U_3 := \texttt{bbba}$, $U_4 := \texttt{bxxx}$, $W := \texttt{abbbab}$, $\alpha_i := \lceil p_i/N \rceil$, and $\beta_i := \lceil (p_i+|W|-1)/N \rceil$ for $i \in \{1, 2\}$. Then, W occurs in $U_1U_2U_3U_4$ at both positions p_1 and p_2, $\beta_1 - \alpha_1 = 1 = \lfloor (|W|-1)/N \rfloor$ and $\beta_2 - \alpha_2 = 2 = \lceil (|W|-1)/N \rceil$.

Lemma 24. *Let f be a polynomial with rational coefficients. The* GPMMS1 *problem remains* NP*-hard even if it is restricted to instances $((S_1, S_2, \ldots, S_m), c, q)$ with $q > f(n)$, where n denotes the length of S_i for any $i \in [\![1, m]\!]$.*

Proof. Each instance $((S_1, S_2, \ldots, S_m), c, q)$ of GPMMS1 is transformed into $((S'_1, S'_2, \ldots, S'_{m'}), c, q')$ where: $m' := m + \lceil |f(n)| \rceil + 1$, $S'_i := S_i$ for each $i \in [\![1, m]\!]$, $S'_i := 1^n$ for each $i \in [\![m+1, m']\!]$, and $q' := q + \lceil |f(n)| \rceil + 1$. As q' is greater than $f(n)$, this transformation only outputs instances of the considered restriction of GPMMS1. Moreover, it is also a Karp-reduction (see Remark 13). Hence, Lemma 7 applies and yields the desired result. □

Lemma 25. *The* GM1 *problem is* NP*-complete.*

Proof. We reduce GPMMS1 to GM1 in order to apply Lemma 7.

Let $((S_1, S_2, \ldots, S_m), c, q)$ be an instance of GPMMS1. Recall that n denotes the non-negative integer satisfying $\{S_1, S_2, \ldots, S_m\} \subseteq \{0, 1\}^n$. According to Lemma 24, we may assume $q > 2N$ where $N := n^3$, and as the case $c \leqslant 1$ is trivial, we may also assume $c \geqslant 2$. Compute a Golomb ruler Γ with m positive marks such that Γ does not contain any pair of consecutive integers. Such a ruler can be obtained by removing a suitable mark from any Golomb ruler with $m+1$ positive marks: for instance, Γ_{m+1} is an $(m+1)$-mark Golomb ruler (see Lemma 17). It can also be shown that Γ_m is a suitable choice for Γ. Let $\gamma_1, \gamma_2, \ldots, \gamma_m$ be an enumeration of Γ, and let M be the greatest element of Γ: $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ and Γ is a subset of $[\![1, M]\!]$. Compute the binary string $T := U_1 U_2 \cdots U_M$ where the N-length strings $U_1, U_2, \ldots, U_M$ are defined by: $U_{\gamma_i} := \widehat{S_i}$ for every $i \in [\![1, m]\!]$, and $U_k := 0^N$ for every $k \in [\![1, M]\!] \setminus \Gamma$. Clearly, (T, c, q) is an instance of GM1 computable from $((S_1, S_2, \ldots, S_m), c, q)$ in polynomial time. It remains to check the next claim.

Claim 26. *$((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of* GPMMS1 *if, and only if, (T, c, q) is a yes-instance of* GM1.

(*only if*) Assume that $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1. This means that there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that index set $I := \{i \in [\![1, m]\!]:\ P \text{ matches } S_i\}$ has cardinality at least q. Apply Lemma 20 with $\mathcal{S} := \{S_i:\ i \in I\}$: there exists a gapped pattern $Q \in \{1, ?\}^\star$ with content c such that Q occurs in $\widehat{S_i} = U_{\gamma_i}$ for every $i \in I$. Hence, there are $\#I \geqslant q$ pairwise non-overlapping occurrences of gapped pattern Q in T: (T, c, q) is a yes-instance of GM1.

(*if*) In order to prove the "if part" of Claim 26, we need two preliminary results.

The next lemma is somehow similar to Lemma 19. It ensures that T contains at most $2N$ occurrences of any proper gapped pattern with length greater than N.

Lemma 27. *Let Q be a gapped pattern with $|Q| > N$ and such that the first and last letters of Q are* 1*s. There exist $i, j \in [\![1, m]\!]$ such that*

$$\mathrm{L}_Q(T) \subseteq [\![(\gamma_i - 1)N + 1, \gamma_i N]\!] \cup [\![(\gamma_j - 1)N + 1, \gamma_j N]\!]. \tag{6}$$

Proof. Let $\delta := (|Q| - 1)/N$ and let A be the set of all $\alpha \in \Gamma$ such that there exists $\beta \in \Gamma$ satisfying $\beta - \alpha = \lfloor \delta \rfloor$ or $\beta - \alpha = \lceil \delta \rceil$. From $|Q| > N$, we deduce $\delta \geqslant 1$, so both integers $\lceil \delta \rceil$ and $\lfloor \delta \rfloor$ are positive. Moreover, Γ is a Golomb ruler, and thus A has cardinality at most two. Pick $i, j \in [\![1, m]\!]$ such that $A = \{\gamma_i, \gamma_j\}$. (Note that A has cardinality less than two whenever δ is an integer, $\lfloor \delta \rfloor \notin \Gamma - \Gamma$, or $\lceil \delta \rceil \notin \Gamma - \Gamma$. In particular, if A has cardinality two then $\lfloor \delta \rfloor$ and $\lceil \delta \rceil$ are two consecutive integers and each of them is measured by a pair of marks in Γ.)

Let us check that Eq. (6) holds. Consider an arbitrary element $p \in \mathrm{L}_Q(T)$. Since $\alpha := \lceil p/N \rceil$ is such that $p \in [\![(\alpha - 1)N + 1, \alpha N]\!]$ (apply Remark 21 with $n := p$), it suffices to prove that $\alpha \in A$. Lemma 22 applies with $W := T[p, p + |Q| - 1]$ and $\beta := \lceil (p + |Q| - 1)/N \rceil$: according to point (i), the first letter and the last letter of Q, which are both 1s, occur in U_α and U_β, respectively. Hence, both U_α and U_β are distinct from 0^N. This requires $\alpha \in \Gamma$ and $\beta \in \Gamma$, respectively. Moreover, according to Lemma 22(iii), $\beta - \alpha = \lfloor \delta \rfloor$ or $\beta - \alpha = \lceil \delta \rceil$. Therefore, we have shown that $\alpha \in A$. This concludes the proof of Lemma 27. □

Gapped patterns with length at most N are handled by the next lemma.

Lemma 28. *Let Q be a gapped pattern with $1 \leqslant |Q| \leqslant N + 1$ and such that the first and last letters of Q are* 1*s. For each $p \in \mathrm{L}_Q(T)$, $\lceil p/N \rceil \in \Gamma$ and Q occurs in $U_{\lceil p/N \rceil}$ at position $p - (\lceil p/N \rceil - 1)N$.*

Proof. Lemma 22 applies with $W := T[p, p + |Q| - 1]$, $\alpha := \lceil p/N \rceil$ and $\beta := \lceil (p + |Q| - 1)/N \rceil$.

From Lemma 22(i) we deduce that the first letter and the last letter of Q, which are both 1s, occur in U_α and U_β, respectively. As in the proof of the previous lemma, it follows that both α and β belong to Γ. Hence, according to Lemma 22(ii), it suffices to check that $\alpha = \beta$ to finish the proof of Lemma 28.

We have $0 \leqslant (|Q| - 1)/N \leqslant 1$, so $0 = \lfloor (|Q| - 1)/N \rfloor \leqslant \lceil (|Q| - 1)/N \rceil \leqslant 1$. Therefore, according to Lemma 22(iii), $\beta - \alpha$ equals zero or one. However, equality $\beta - \alpha = 1$ cannot happen since otherwise α and β would be two consecutive integers belonging to Γ. We have thus shown $\alpha = \beta$. This concludes the proof of Lemma 28. □

Assume that (T, c, q) is a yes-instance of GM1. This means that there exists a gapped pattern $Q \in \{1, ?\}^\star$ with content c such that $|T|_Q \geqslant q$. Let I be the set of all indices $i \in [\![1, m]\!]$ such that Q occurs in $\widehat{S_i}$. Lemma 20 applies with $\mathcal{S} := \{S_i\colon\ i \in I\}$: there exists a gapped pattern $P \in \{1, ?\}^n$ with content c such that for every $i \in I$, P matches S_i. Hence, to show that $((S_1, S_2, \ldots, S_m), c, q)$ is a yes-instance of GPMMS1, it suffices to check that the cardinality of I is greater than or equal to $|T|_Q$. Roughly speaking, we check that any two distinct occurrences of Q in T are contained in two distinct substrings $\widehat{S_i}$. Without loss of generality, we may assume that the first and last letters of Q are 1s. As we also assume $q > 2N$, Lemma 27 ensures $|Q| \leqslant N$. Hence, Lemma 28 applies: for each $p \in \mathrm{L}_Q(T)$, we may pick $i_p \in [\![1, m]\!]$ such that $\gamma_{i_p} = \lceil p/N \rceil$; moreover Q occurs in $U_{\lceil p/N \rceil} = \widehat{S_{i_p}}$ at position $p - (\gamma_{i_p} - 1)N$, and thus $i_p \in I$. Let us check that the mapping $p \in \mathrm{L}_Q(T) \mapsto i_p \in I$ is injective. Let $x, y \in \mathrm{L}_Q(T)$ with $x \neq y$. If we had $i_x = i_y$ then Q would occur in $\widehat{S_{i_x}} = \widehat{S_{i_y}}$ at two distinct positions, namely $x - (\gamma_{i_x} - 1)N$ and $y - (\gamma_{i_y} - 1)N$: contradiction with Lemma 19. This concludes the proof of Lemma 25. □

4.2. Reduction from GM1 to GM

In order to prove that GM1 Karp-reduces to GM, we need three simple lemmas.

Lemma 29. *For every integer $C \geqslant 0$ and every gapped pattern $Q \in \{1, ?\}^\star$, $|1^C|_Q \geqslant C - |Q| + 1$.*

Proof. If $|Q| \leqslant C$ then $\mathrm{L}_Q(1^C) = [\![1, C - |Q| + 1]\!]$ and thus $|1^C|_Q = C - |Q| + 1$. If $|Q| > C$ then Q is too long to occur in 1^C and thus $|1^C|_Q = 0$. □

Lemma 30. *Let W and S be two strings and let Q be a gapped pattern. If Q occurs in WS at more than $|W|$ distinct positions then Q occurs in S.*

Proof. The cardinality of $\mathrm{L}_Q(WS)$ is greater than the cardinality of $[\![1, |W|]\!]$, so $\mathrm{L}_Q(WS)$ is not a subset of $[\![1, |W|]\!]$. Therefore, there exists an element $p \in \mathrm{L}_Q(WS)$ with $p \notin [\![1, |W|]\!]$. Clearly, Q occurs in S at position $p - |W|$. □

Lemma 31. *Let X and Y be two strings, let B be a non-negative integer, and let Q be a non-empty gapped pattern. If $|Q| \leqslant B + 1$ and if neither the first nor the last letter of Q matches 0 then $|X0^BY|_Q = |X|_Q + |Y|_Q$.*

Proof. Inequality $|X0^BY|_Q \geqslant |X|_Q + |Y|_Q$ is clear. The converse inequality is demonstrated as follows: given an occurrence of Q in $X0^BY$, we check that it is fully contained either in prefix X or in suffix Y. Since the length of Q is at most $B + 1$, the considered occurrence is fully contained in prefix $X0^B$ or in suffix 0^BY of $X0^BY$. However, the last letter of Q does not match 0 (i.e., $Q[|Q|] \notin \{?, 0\}$) and thus any occurrence of Q in $X0^B$ is fully contained in its prefix X. In the same way, since the first letter of Q does not match 0 (i.e., $Q[1] \notin \{?, 0\}$), any occurrence of Q in 0^BY is fully contained in its suffix Y. We have thus shown $|X0^BY|_Q = |X|_Q + |Y|_Q$. □

Theorem 32. *The* GM *problem is* NP*-complete, even if the input alphabet is binary.*

Proof. We reduce GM1 to GM in order to apply Lemma 25.

Let (T, c, q) be an instance of GM1. Compute the following four integers: $A := |T|$, $B := |T|^2 + (1-q)|T|$, $C := |T|^3 + (2-q)|T|^2 + (1-q)|T|$, and $q' := |T|^3 + (2-q)|T|^2 + 1$. If $q \geqslant |T| + 2$ then (T, c, q) is a no-instance of GM1, and if $q = 0$ then (T, c, q) is a yes-instance of GM1. Hence, we may assume $1 \leqslant q \leqslant |T| + 1$ without loss of generality. In particular, inequality $q \leqslant |T| + 1$ ensures that A, B, C and q' are non-negative. This allows us to construct

$$T' := \left(T0^B\right)^A 1^C.$$

To prove Theorem 32, it is sufficient to check that the transformation $(T, c, q) \mapsto (T', c, q')$ induces a Karp-reduction from GM1 to GM. First, (T', c, q') is clearly computable in polynomial time from (T, c, q). Thus, it remains to show that (T, c, q) is a yes-instance of GM1 iff (T', c, q') is a yes-instance of GM. It suffices to check the next claim.

Claim 33. *Let Q be a gapped pattern with non-zero content and such that neither the first nor the last letter of Q is a ?. Both $Q \in \{1, ?\}^\star$ and $|T|_Q \geqslant q$ hold iff $|T'|_Q \geqslant q'$.*

It is easy to see that A, B, C and q' satisfy the following four inequalities:

$$q' \leqslant qA + C - |T| + 1, \tag{7}$$

$$q' \geqslant \big(|T| + B\big)A + 1, \tag{8}$$

$$q' \geqslant A|T| + C - B, \tag{9}$$

$$q' \geqslant (q-1)A + C + 1. \tag{10}$$

In fact, the five integers $qA + C - |T| + 1$, q', $(|T| + B)A + 1$, $A|T| + C - B + 1$, and $(q-1)A + C + 1$ are equal. Inequality (7) is used to prove the "only if part" of Claim 33 while its "if part" is deduced from inequalities (8), (9) and (10).

In order to ease notation, the prefix $(T0^B)^A$ of T' is denoted W:

$$T' = W1^C.$$

We can now turn to the proof of Claim 33.

(*only if*) Assume $Q \in \{1, ?\}^\star$ and $|T|_Q \geqslant q$. On the one hand, Lemma 29 ensures $|1^C|_Q \geqslant C - |Q| + 1$. Moreover, Q occurs in T since we assume $q \geqslant 1$, and thus Q cannot be longer than T: $|1^C|_Q \geqslant C - |T| + 1$. On the other hand, there are A pairwise non-overlapping occurrences of T in W, and by hypothesis, Q occurs in each at q distinct positions or more: $|W|_Q \geqslant A|T|_Q \geqslant qA$. Summing up, we obtain

$$\left|T'\right|_Q \geqslant |W|_Q + \left|1^C\right|_Q \geqslant qA + C - |T| + 1,$$

and according to inequality (7), the latter integer is at least q'. We have thus shown $|T'|_Q \geqslant q'$.

(*if*) Conversely, assume $|T'|_Q \geqslant q'$. Inequality (8) is equivalent to $q' \geqslant |W| + 1$, and thus there are at least $|W| + 1$ occurrences of Q in $W1^C = T'$. Hence, Lemma 30 applies with $S := 1^C$: Q occurs in 1^C. Therefore, Q is an element of $\{1, ?\}^\star$ and the length of Q is at most C (see Remark 5). Let us check that $|Q| \leqslant B + 1$. The first letter of Q is a 1, and thus $T'[p] = 1$ for every $p \in \mathrm{L}_Q(T')$. Hence, $|T'|_Q$ is bounded from above with the number of letters 1 occurring in the $(|T'| - |Q| + 1)$-length prefix of T'. Since the length of Q is at most C, the latter prefix equals $W1^{C-|Q|+1}$. We can now write

$$q' \leqslant \left|T'\right|_Q \leqslant \left|W1^{C-|Q|+1}\right|_1 = A|T|_1 + C - |Q| + 1 \leqslant A|T| + C - |Q| + 1,$$

and then inequality (9) yields $A|T| + C - B \leqslant A|T| + C - |Q| + 1$, which is equivalent to $|Q| \leqslant B + 1$. Now, applying A times Lemma 31 yields $|T'|_Q = A|T|_Q + |1^C|_Q$. Besides, we have $|1^C|_Q \leqslant C$ because Q is non-empty. It follows

$$A|T|_Q + C \geqslant \left|T'\right|_Q \geqslant q' \geqslant (q-1)A + C + 1$$

by inequality (10). This requires $|T|_Q \geqslant q$. □

5. A quadratic-time algorithm for the Gapped Tandem Repeat problem

We first present a naive cubic-time algorithm for GTR, and then improve it into a quadratic-time algorithm. The starting point is the following basic remark:

Remark 34. Let S be a string and let q be an integer such that q divides the length of S. There exists a unique gapped pattern Q with maximum content, among all gapped patterns whose qth powers match S; Q has length $\ell := |S|/q$ and for each index $i \in [\![1, \ell]\!]$, the ith letter of Q is given by

- $Q[i] = S[i]$ if $S[i] = S[i + \ell] = S[i + 2\ell] = \cdots = S[i + (q-1)\ell]$, and
- $Q[i] = ?$ otherwise.

Hence, each of the ℓ letters of Q is computable in $O(q)$ time, and thus Q is computable from S and q in $O(|S|)$ time.

Let T be a string and let q be a positive integer: (T, q) is an arbitrary instance of GTR. Note that if a non-empty gapped pattern Q is such that Q^q occurs in T, then the length of Q is an element of $[\![1, \lfloor |T|/q \rfloor]\!]$.

Lemma 35. *For each $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ and each $p \in [\![1, |T| - q\ell + 1]\!]$, there exists a unique gapped pattern with maximum content, among all ℓ-length gapped patterns whose qth powers occur in T at position p; it is denoted $Q_{\ell,p}$; $Q_{\ell,p}$ is computable from ℓ, p, q and T in $O(|T|)$ time.*

Proof. Apply Remark 34 with $S := T[p, p + q\ell - 1]$. □

A simple way to solve GTR on input (T, q) is to compute all gapped patterns of the form $Q_{\ell,p}$ with $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ and $p \in [\![1, |T| - q\ell + 1]\!]$. This procedure can be achieved in cubic time $O(|T|^3/q)$: $\Theta(|T|^2/q)$ gapped patterns are to be computed, and by Lemma 35, each pattern is computable in $O(|T|)$ time. We now present an algorithm that solves GTR on any input (T, q) in quadratic time $O(|T|^2)$.

Definition 36. For each $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ and each $p \in [\![1, |T| - q\ell + 1]\!]$, let $c_\ell(p)$ denote the maximum content, over all ℓ-length gapped patterns Q such that Q^q occurs in T at position p: $c_\ell(p)$ is the content of $Q_{\ell,p}$.

Algorithm 1: An $O(|T|^2)$ time algorithm for GTR.

Input: A string T and a positive integer q.
Output: A gapped pattern with maximum content, among all gapped patterns Q such that Q^q occurs in T.
$c^* := -1$;
for $\ell := 1$ **to** $\lfloor |T|/q \rfloor$ **do**
  Compute the maximum content, over all ℓ-length gapped patterns Q such that Q^q occurs in T at position 1, and store this value in variable c ;
  for $p := 1$ **to** $|T| - q\ell$ **do**
    if $c > c^*$ **then**
      $c^* := c$;
      $\ell^* := \ell$;
      $p^* := p$;
    if *the letters* $T[p + k\ell]$ *with* $k \in [\![0, q-1]\!]$ *are all equal* **then**
      $c := c - 1$;
    if *the letters* $T[p + k\ell]$ *with* $k \in [\![1, q]\!]$ *are all equal* **then**
      $c := c + 1$;
    Invariant: at this point, the value of variable c equals $c_\ell(p+1)$.
  if $c > c^*$ **then**
    $c^* := c$;
    $\ell^* := \ell$;
    $p^* := p$;
Compute the ℓ^*-length pattern Q with content c^* such that Q^q occurs in T at position p^* ;
return Q ;

Let c^* denote the maximum content, over all gapped patterns Q such that Q^q occurs in T. The algorithm proceeds as follows. First, all integers of the form $c_\ell(p)$, with $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ and $p \in [\![1, |T| - q\ell + 1]\!]$, are computed, leading to the identification of a pair (ℓ^*, p^*) such that $c^* = c_{\ell^*}(p^*)$. This step is achieved in $O(|T|^2)$ time as explained below. Then, relying on Lemma 35, Q_{ℓ^*,p^*} is computed and returned without increasing the asymptotic running time.

It remains to prove the quadratic-time complexity bound for the computation of all $c_\ell(p)$s.

Lemma 37. *For every* $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ *and every* $p \in [\![1, |T| - q\ell]\!]$,

$$c_\ell(p+1) = c_\ell(p) - \chi_\ell(p) + \chi_\ell(p+\ell),$$

where for every $i \in [\![1, |T| - (q-1)\ell]\!]$, *the indicator* $\chi_\ell(i)$ *is defined by*:

- $\chi_\ell(i) := 1$ *if* $T[i] = T[i+\ell] = T[i+2\ell] = \cdots = T[i+(q-1)\ell]$, *and*
- $\chi_\ell(i) := 0$ *otherwise.*

Proof. It is easy to deduce from Remark 34 that

$$c_\ell(p) = \sum_{i=p}^{p+\ell-1} \chi_\ell(i)$$

for every $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$ and every $p \in [\![1, |T| - q\ell + 1]\!]$. Lemma 37 follows. □

Proposition 38. *For each* $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$, *the* $(|T| - q\ell + 1)$*-tuple of integers* $(c_\ell(1), c_\ell(2), \ldots, c_\ell(|T| - q\ell + 1))$ *is computable from* ℓ, q *and* T *in* $O(q|T|)$ *time.*

Proof. According to Lemma 35, $Q_{\ell,1}$ and its content $c_\ell(1)$ are computable in $O(|T|)$ time. Moreover, for each $p \in [\![1, |T| - q\ell]\!]$, $c_\ell(p+1)$ is computable from $c_\ell(p)$ in $O(q)$ time using the recurrence relation stated in Lemma 37: evaluating $\chi_\ell(p)$ and $\chi_\ell(p+\ell)$ takes $O(q)$ time. Hence, $(c_\ell(1), c_\ell(2), \ldots, c_\ell(|T| - q\ell + 1))$ is computable in $O(|T| + q \times (|T| - q\ell)) = O(q|T|)$ time. □

Computing all $c_\ell(p)$s is computing the $\lfloor |T|/q \rfloor$ tuples of the form $(c_\ell(1), c_\ell(2), \ldots, c_\ell(|T| - q\ell + 1))$ with $\ell \in [\![1, \lfloor |T|/q \rfloor]\!]$. According to Proposition 38, this can be achieved in $O(\lfloor |T|/q \rfloor \times q|T|) = O(|T|^2)$ time, as claimed.

Summarizing the preceding discussion, we obtain Algorithm 1.

Acknowledgement

This work is supported by the Academy of Finland under grant 7523004 (Algorithmic Data Analysis).

References

[1] A. Apostolico, L. Parida, Incremental paradigms of motif discovery, J. of Computational Biology 11 (1) (2004) 15–25.
[2] H. Arimura, T. Uno, An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence, J. of Combinatorial Optimization 13 (3) (2007) 243–262.
[3] W.C. Babcock, Intermodulation interference in radio systems, Bell System Technical Journal 32 (1) (1953) 63–73.
[4] M. Crochemore, C.S. Iliopoulos, M. Mohamed, M.-F. Sagot, Longest repeats with a block of k don't cares, Theoretical Computer Science 362 (1–3) (2006) 248–254.
[5] G. Das, R. Fleischer, L. Gąsieniec, D. Gunopulos, J. Kärkkäinen, Episode matching, in: A. Apostolico, J. Hein (Eds.), Proc. of the 8th Annual Symposium on Combinatorial Pattern Matching, CPM'1997, in: LNCS, vol. 1264, Springer-Verlag, 1997, pp. 12–27.
[6] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Computer Science and Computational Biology, Cambridge University Press, 1997.
[7] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, 1972, pp. 85–103.
[8] R.M. Kolpakov, G. Kucherov, Finding maximal repetitions in a word in linear time, in: Proc. of the 40th Annual Symposium on Foundations of Computer Science, FOCS'1999, IEEE Computer Society, 1999, pp. 596–604.
[9] R.M. Kolpakov, G. Kucherov, Finding approximate repetitions under Hamming distance, Theoretical Computer Science 303 (1) (2003) 135–156.
[10] J. Kevin Lanctot, M. Li, B. Ma, S. Wang, L. Zhang, Distinguishing string selection problems, Information and Computation 185 (1) (2003) 41–55.
[11] G.M. Landau, J.P. Schmidt, D. Sokol, An algorithm for approximate tandem repeats, J. of Computational Biology 8 (1) (2001) 1–18.
[12] M. Li, B. Ma, L. Wang, Finding similar regions in many sequences, J. of Computer and System Sciences 65 (1) (2002) 73–96.
[13] L. Marsan, M.-F. Sagot, Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification, J. of Computational Biology 7 (3–4) (2000) 345–362.
[14] F. Nicolas, É. Rivals, Hardness of optimal spaced seed design, J. of Computer and System Sciences 74 (5) (2008) 831–849.
[15] L. Parida, I. Rigoutsos, A. Floratos, D.E. Platt, Y. Gao, Pattern discovery on character sets and real-valued data: linear bound on irredundant motifs and an efficient polynomial time algorithm, in: Proc. of the 11th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA'2000, 2000, pp. 297–308.
[16] J. Pelfrêne, S. Abdeddaïm, J. Alexandre, Extracting approximate patterns, J. of Discrete Algorithms 3 (2–4) (2005) 293–320.
[17] N. Pisanti, M. Crochemore, R. Grossi, M.-F. Sagot, Bases of motifs for generating repeated patterns with wild cards, IEEE/ACM Transactions on Computational Biology and Bioinformatics 2 (1) (2005) 40–49.
[18] E. Rocke, Using suffix trees for gapped motif discovery, in: R. Giancarlo, D. Sankoff (Eds.), Proc. of the 11th Annual Symposium on Combinatorial Pattern Matching, CPM'2000, in: LNCS, vol. 1848, Springer-Verlag, 2000, pp. 335–349.
[19] J.A. Storer, Data Compression: Methods and Theory, Principles of Computer Science, vol. 13, Computer Science Press, 1988.